

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

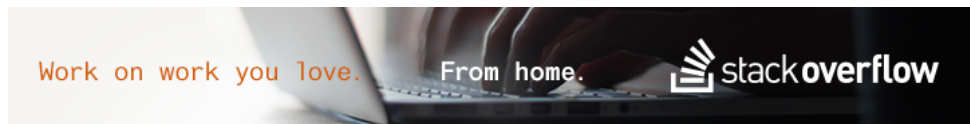
Join the Stack Overflow community to:

Ask
programming
questions

Answer and help
your peers

Get recognized for your
expertise

Get from AnyObject(NSString) to String



I am reading a plist key (NSArray with n NSDictionary):

```
let regionsToMonitor = NSBundle.mainBundle().infoDictionary["Regions"] as
Array<Dictionary<String, AnyObject>>
```

now I iterate over it:

```
for regionToMonitor in regionsToMonitor {
```

and now I want to get uuidString of the regionToMonitor

```
in ObjC: NSString *uuidString = regionToMonitor[@"uuidString"];
```

```
in swift I try: let uuidString = regionToMonitor["uuid"]!.stringValue;
```

the above does compile but the string is always nil in swift. regionToMonitor["uuid"] when used without !.stringValue works fine in println

how do I get a valid Swift.String here?

I am trying to pass it to NSUUID!

I also tried

```
let uuidString:String = regionToMonitor["uuid"]
=> AnyObject isn't convertible to String
```

```
let uuidString = regionToMonitor["uuid"] as String
=> Could not find an overload for 'subscript' that accepts the supplied arguments
```

```
let uuidString = regionToMonitor["uuid"];
=> 'AnyObject?' cannot be implicitly downcast to 'String'; did you mean to use 'as' to force downcast?
```

ios nsstring swift

edited Jun 9 '14 at 7:03



fabian789
4,937 3 30 73

asked Jun 3 '14 at 0:20



Daij-Djan
32.6k 11 48 95

NSString == String in Swift. – Leandros Jun 3 '14 at 0:29

thanks thats what the IDE says too ;) - thats what I thought but this doesn't work -- see my EDIT – Daij-Djan Jun 3 '14 at 0:32

7 Answers

I ended up with the ugly line:

```
var uuidString:String = regionToMonitor["uuid"] as! String
```

no warnings, no errors, no runtime error

edited May 20 '15 at 19:42



John Erck
5,013 2 35 57

answered Jun 3 '14 at 6:23



Daij-Djan
32.6k 11 48 95

+1.nice, sort and sweet. – Renish Dadhaniya Jul 18 '14 at 11:15

Why do you need to type uuidString, won't it be the same if you do `var uuidString = regionToMonitor["uuid"] as String!` – [Gerard Diaz](#) Oct 29 '14 at 14:46

in the beta I developed this for, not typing it CRASHED :) – [Daij-Djan](#) Oct 29 '14 at 14:47

1 By the way, if `regionToMonitor` were an optional `NSDictionary` or `Dictionary`, Xcode 6.3 would say that `AnyObject?` cannot be cast to `String`, which is misleading. – [sudo](#) Jun 10 '15 at 19:36

@sudo Your comment got me out of 20 minute error search. Indeed, Xcode is reporting totally unrelated error, I had to add `regionToMonitor!["blah"]` and tadaaaaa....no errors :) – [Igor Pantović](#) Aug 19 '15 at 18:31

Add  projects to your  **stackoverflow** profile.

I found this to work for me

```
var uuidString: String? = regionToMonitor["uuid"] as AnyObject? as? String
```

edited Aug 7 '14 at 12:38

answered Jul 20 '14 at 9:15



[Amitay](#)

413 3 18

this worked but it feels gross... thanks @Amitay – [Dan Beaulieu](#) Apr 9 '15 at 21:17

`AnyObject?` is an optional, because the dictionary may or may not contain a value for the "uuid" key. To get at an optional's value, you have to *unwrap* it. See [Optionals](#) in the documentation.

The safest way to deal with an optional is to put it in a conditional statement.

```
if let uuidString = regionToMonitor["uuid"] {
    // do something with uuidString
}
```

If you're absolutely positively sure the dictionary will always contain this key/value pair, you can use an implicitly unwrapped optional (the `!` suffix):

```
println("UUID: \(regionToMonitor["uuid"]!)" )
```

In this case, if there's no value for the key your app will crash.

If you use `!` a lot, it looks like you're yelling all the time... which might help illustrate why you should use it sparingly, if at all. :)

edited Jun 3 '14 at 1:11

answered Jun 3 '14 at 0:56



[rickster](#)

55.7k 9 103 150

Try removing some of the extra downcasting (`as ...`) you're doing before accessing the dictionary - this might be an issue with inferred types. – [rickster](#) Jun 3 '14 at 1:04

I'm not completely sure about, what the `NSBundle` returns here. The first method does indeed work (but not without a warning), the second doesn't (not just because you're missing a parenthesis). – [Leandros](#) Jun 3 '14 at 1:08

Cmd-click `NSBundle` in your code to see its Swift declarations. (Xcode automatically Swift-ifies the header on demand, with comments.) That might help track down type issues. – [rickster](#) Jun 3 '14 at 1:11

As for missing parens... clearly someone needs to figure out how to embed SO in a playground or vice versa. :D – [rickster](#) Jun 3 '14 at 1:11

`var regions = NSBundle mainBundle().infoDictionary["Regions"]` is `AnyObject?`, but it contains an `NSArray` with zero entries. It definitely doesn't work without downcast. – [Leandros](#) Jun 3 '14 at 1:12

I've found a working solution, which compiles without warnings and such:

```
var regions = NSBundle mainBundle().infoDictionary["Regions"] as Array<Dictionary<String, AnyObject>>

for region in regions {
    let dict: NSDictionary = region
    var uuid = dict["uuidString"] as String
}
```

The `infoDictionary` from the `NSBundle` returns an `NSArray` and `NSDictionary`, not a `Swift.Array` or `Swift.Dictionary`. Though, they should be interchangeable, but maybe they aren't as we thought.

answered Jun 3 '14 at 1:22

1



Leandros

10.6k 5 42 74

let dict: NSDictionary = region did the trick for me – Daij-Djan Jun 3 '14 at 6:07

don't think that is cool but that works fine – Daij-Djan Jun 3 '14 at 6:12

correction: if do this + a println -- the whole thing comes crashing don EXC_INVOP on reading the plist – Daij-Djan Jun 3 '14 at 6:20

I am not sure my solution is effective of not but here it is.

```
var uuidVar = regionToMonitor["uuid"]
var uuidString:String = "\\(uuidVar)"
```

Hope it helps.

answered Apr 7 '15 at 11:41



ACengiz

804 12 12

You can also use

```
var uuidString = regionToMonitor["uuid"]? as String
```

It has the same results as what you are doing, but is IMHO more clear in intent. The `as` operator force unwraps anyway, so putting the exclamation mark behind it feels redundant. Putting the question mark behind the dictionary subscript makes it clear you are chaining an optional.

answered Jun 10 '14 at 9:08



Mike Seghers

465 1 6 10

Doing this causes my swift process to crash with error 254. See stackoverflow.com/questions/24154163/... for details. – tng Jul 5 '14 at 4:50

If you are sure you want the unwrapped value you can use any of these:

```
var uuidString:String! = regionToMonitor["uuid"]
var uuidString = regionToMonitor["uuid"] as String!
```

or even this:

```
if var uuidString = regionToMonitor["uuid"] {
    println("\\(uuidString) has been unwrapped")
}
```

answered Aug 15 '14 at 23:08



ggomez

3,823 4 20 30